

Teaching guide: Relational operations

This resource will help with understanding relational operations. It supports Section 3.2.4 of our current GCSE Computer Science specification (8520). The guide is designed to address the following learning aims:

- Recognise the six relational operators.
- Evaluate Boolean expressions that use relational operators.

Ordering integers

Many data types have a logical order, if you can place one value higher or lower than another then generally speaking you compare them using the following six operations shown in the table. Some types such as Booleans cannot be ordered but they can be said to be equal or unequal (i.e. `True > False` does not make sense, but `True ≠ False` does).

Logical Operator	Meaning	Example
<	less than	<code>3 < 4</code> means 3 is less than 4
>	greater than	<code>4 > 3</code> means 4 is greater than 3
=	equal to	<code>3 = 3</code> means 3 is equal to 3
≠	not equal to	<code>3 ≠ 4</code> means 3 is not equal to 4
≤	less than or equal to	<code>3 ≤ 4</code> means 3 is less than or equal to 4
≥	greater than or equal to	<code>4 ≥ 4</code> means 4 is greater than or equal to 4

All of the example expressions in the table evaluate to `true`. In fact, all expressions with relational operations evaluate to either `true` or `false`, that is, a Boolean value.

Arithmetic operations are evaluated before relational operations and it is fairly simple to combine the two. For example:

$(3 * 4) \leq 15$ evaluates to
 $12 \leq 15$ which evaluates to
True

$(15 \text{ MOD } 2) = 0$ evaluates to
 $1 = 0$ which evaluates to
False

$(500 - 400) > (3 * (2 + 5))$ evaluates to
 $100 > 21$ which evaluates to
True

Ordering other types

All real numbers can be compared using the six relational operators above. Strings can also be ordered using 'dictionary' or lexicographic order, e.g. 'aardvark' is less than 'apple' because it comes before it in the dictionary. There are many other types that can be ordered in this way although, once again, it is dependent on the language used. For instance, you would expect yesterday's date to be less than tomorrow's date but you will have to check whether your language allows you to do this.